

Hoitoaikojen ilmoituspalvelu päiväkodeille ja vanhemmille

Henri Komulainen



Tekijä(t) Komulainen Henri	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Hoitoaikojen ilmoituspalvelu päiväkodeille ja vanhemmille	Sivu- ja liitesivumäärä 27 + 1
Opinnäytetyön otsikko englanniksi Child care times notification service for kindergartens and parents	
<p>Opinnäytetyön tarkoituksena oli toteuttaa Hoitoajat-verkkopalvelu päiväkotien ja sen asiakkaiden eli vanhempien käyttöön. Verkkopalvelun kautta vanhemmat pystyivät ilmoittamaan lasten hoitoaikoja. Päiväkodin henkilökunta pystyi tarkastelemaan näitä ilmoitettuja hoitoaikoja verkkopalvelun kautta sekä hyväksymään tai hylkäämään ilmoitukset. Päiväkotien pystyi suunnittelemaan työvuoronsa ilmoitettujen hoitoaikojen perusteella.</p> <p>Opinnäytetyössä toteutettiin yksinkertainen Hoitoajat-verkkopalvelun prototyyppi ja se toimivuus testattiin, jotta se vastasi sille laadittuihin vaatimuksiin. Verkkopalvelun vaatimuksia olivat skaalautuvuus näyttökoon mukaan, helppokäyttöisyys, tietoturvallisuus sekä laajennettavuus. Verkkopalvelu toteutettiin Wordpress-sisällönhallintajärjestelmällä. Opinnäytetyön tietoperustassa käytiin läpi ohjelmistokehityksen käytäntöjä sekä yleistä asiaa sisällönhallintajärjestelmistä.</p> <p>Hoitoajat-verkkopalvelun toteuttaminen osoitti, että verkkopalveluita on helppo luoda kunhan vain ymmärtää ne tarpeet joihin sen tulisi vastata sekä riittävä ymmärrys sovelluskehityksestä. Sisällönhallintajärjestelmänä Wordpress osoittautui hyvin käyttäjäystävälliseksi ja totutun jälkeen sillä sai aikaan toimivan verkkopalvelun. Jatkotutkimusmahdollisuuksina näen verkkopalvelun lisäarvon kasvattamisen jatkokehityksellä sekä verkkopalvelun tuotteistamisen.</p>	
Asiasanat Ilmoitus, verkkopalvelu, päiväkodit, HTML, PHP, MySQL, Wordpress, sisällönhallintajärjestelmä	

Author(s) Komulainen Henri	
Degree programme Business Information Technology	
Report/thesis title Child care times notification service for kindergartens and parents	Number of pages and appendix pages 27 + 1
<p>The purpose of this thesis was to implement a child care times notification service for kindergartens and to their customers, the parents. The parents must inform the kindergarten about the care times; when the child will be at the kindergarten. The parents were able to send the child care times through the webservice called Hoitoajat. The kindergarten staff were able to examine these child care times via the Hoitoajat-webservice, as well as to approve or reject them. The kindergarten could plan its workshifts according to the child care times that were sent to them.</p> <p>A simple Hoitoajat-web service was implemented in the thesis and it was tested so that it worked as intended. There were requirements for the web service such as scalability according to screen size, ease of use, security, and expandability. The web service was created with WordPress content management system. The theory part of the thesis consists of the software development practices, as well as general information about the content management systems.</p> <p>The implementation of Child care times-web service showed that web services are easy to create as long as one understands the needs to which it should respond and bear sufficient understanding of software development practices. Content management system WordPress turned out to be very user friendly. After getting used to it, I could create a working web service. For future thesis research opportunities I see adding value to the web service by further developing it and commercialization of the web service.</p>	
Keywords Notification, webservice, kindergarten, HTML, PHP, MySQL, Wordpress, Content management system	

Sisällys

1	Johdanto	1
1.1	Opinnäytetyön tavoite ja rajausta	1
1.2	Sanasto ja käsitteet	2
2	Tietoperusta	3
2.1	Verkkopalvelun määritelmä	3
2.2	Käyttöliittymän suunnittelu	4
2.3	Konseptin suunnittelu	5
2.3.1	Toimintatarinat	6
2.3.2	Vaatimusmäärittely ja skenaariot	6
2.3.3	Käyttötarinat	7
2.4	Testaaminen ja testauksen suunnittelu	8
2.4.1	Yksikkötestaus	8
2.4.2	Integrointitestaus	8
2.4.3	Järjestelmätestaus	9
2.5	Sisällönhallintajärjestelmät	9
2.5.1	Wordpress	10
3	Hoitoajat-verkkopalvelun vaatimusmäärittely	11
3.1	Verkkopalvelun yleiset vaatimukset	11
3.1.1	Toimintatarina nykytilasta	12
3.1.2	Hoitoajat-verkkopalvelun skenaariot	12
3.1.3	Käyttötarinat	13
4	Verkkopalvelun luominen sisällönhallintajärjestelmällä	15
4.1	Hoitoajat-verkkopalvelu	15
4.2	Hoitoajat-verkkopalvelun toiminnot	16
4.2.1	Sisäänkirjautuminen	16
4.2.2	Hoitoaikojen ilmoittaminen	17
4.2.3	Omien tietojen muokkaus	17
4.2.4	Käyttäjien rekisteröinti	17
4.2.5	Käyttäjien hallinnointi	17
4.2.6	Ilmoitettujen hoitoaikojen käsittely	18
4.2.7	Ilmoitettujen ja hyväksytyjen hoitoaikojen tarkastelu	18
5	Hoitoajat-verkkopalvelu toteutus	18
5.1	Käytetyt lisäosat	18
5.2	Sisäänkirjautumissivu	19
5.3	Unohtunut salasana	19
5.4	Käyttäjän etusivu	20
5.5	Kuittaus lähetetyistä hoitoajoista	21
5.6	Käyttäjän tiedot-sivu	21

5.7	Pääkäyttäjän etusivu	22
5.8	Pääkäyttäjän kalenteri näkymä	23
5.9	Pääkäyttäjän näkymä käyttäjiin	23
5.10	Uuden käyttäjän lisääminen	24
6	Testaaminen	25
6.1	Yksikkötestaus	25
6.2	Integroititestaus.....	25
6.2.1	Testitapaukset integroititestaukselle	25
6.3	Järjestelmätestaus	26
	Pohdinta	27
	Lähteet	28

1 Johdanto

Hoitoaikojen ilmoituspalvelu päiväkodeille ja vanhemmille on tarpeesta syntynyt konsepti. Tarkoituksena on luoda verkkopalvelun prototyyppi, jossa vanhemmat voivat ilmoittaa lapsen hoitoajat päiväkodille jotta päiväkotit voi suunnitella työvuoronsa ja päivän aktiviteetit. Verkkopalveluun kirjaudutaan sisään ja vanhemmilla sekä päiväkodilla on omat tunnukset. Verkkopalvelun prototyypissä vanhemmat voivat ilmoittaa lapsen hoitoajat sisään kirjautumisen jälkeen. Tieto tallentuu verkkopalveluun, josta päiväkotit voi nähdä kaikkien lasten hoitoajat päiväkohtaisesti.

Aihe syntyi omakohtaisesta kokemuksesta. Seuraavan viikon hoitoajat pitää ilmoittaa etukäteen paperilapulla edellisen viikon keskiviikkoon mennessä. Unohdin kerran toimittaa lapun keskiviikko aamuna ja totesin itselleni, että miksi tätä ei ole tehty verkkopalveluksi jotta ilmoitukset voisi tehdä verkon kautta.

Opinnäytetyön keskeisinä kysymyksinä olivat:

- Miten ilmoituspalvelu toteutetaan responsiivisena verkkopalveluna?
- Miten tunnusten luominen ja sisäänkirjautuminen toteutetaan verkkopalvelulle?
- Miten lomakkeelle syötetyt tiedot tallennetaan verkkopalveluun ja saadaan näky-mään päiväkodin henkilökunnalle?

Lopullisena tuotoksena siis syntyi verkkopalvelun prototyyppi, jota voi käyttää ja tarkastella millä tahansa verkkoon kytketyllä laitteella jossa on verkkoselain. Tästä verkkopalvelusta hyötyy sekä pienten lasten vanhemmat ja päiväkodit. Vastaavaan palveluun en ole aiemmin törmännyt.

1.1 Opinnäytetyön tavoite ja rajaus

Tavoitteena oli saada aikaan toimiva prototyyppi verkkopalvelusta, jossa vanhemmat voivat ilmoittaa lapsensa hoitoajat ja päiväkotit voi nähdä ilmoitetut hoitoajat. Opinnäytetyötä ei tehty toimeksiantona kenellekään.

Oletuksena oli, että lukijalla on perusymmärrys HTML5 - ja CSS - ohjelmointikielistä sekä ohjelmistokehityksen perusteista. Opinnäytetyön tarkoituksena ei ole opettaa tai perehdyttää lukijaa ohjelmoinnin perusteisiin vaan toimia käytännön esimerkkinä kuinka perusteita soveltamalla voidaan luoda verkkopalvelun prototyyppi.

Prototyyppi rajataan niin, että vain seuraavat toiminnallisuudet ovat osana lopullista prototyyppiä:

- Vanhemmat voivat ilmoittaa vain yhden lapsen hoitoajat eli käyttäjätunnus per lapsi
- Päiväkoti voi nähdä ilmoitetut hoitoajat listana, josta käy ilmi päivämäärä, lapsen nimi, tuloaika ja noutoaika
- Verkkopalvelu on responsiivinen eli sivustoa voi katsella millä tahansa laitteella ja sivusto skaalautuu laitteen näyttöpäätteen mukaan

1.2 Sanasto ja käsitteet

CSS	Cascading Style Sheets eli CSS kokoaa yhteen tyyliohjeet, joiden perusteella verkkosivut esitetään halutulla tavalla. CSS-tiedosto antaa verkkosivulle sen värit, muodot ja rakenteen.
HTML5	HTML5 on HTML-merkintäkielen uusin versio, joka toimii pohjana monille nykyaikaisille web-tekniikoille. HTML5:llä voidaan esittää verkkosivun sisältö halutulla tavalla.
PHP	On ohjelmointikieli, jolla voidaan luoda dynaamisia web-sivuja ja web-palveluita. PHP on joustava ohjelmointikieli ja sitä voidaan käyttää useilla eri alustoilla ja käyttöjärjestelmillä.
MySQL	Hyvin suosittu ja ilmainen relaatiotietokantaohjelmisto.
Ohjelmointikieli	Tämän opinnäytetyön aikana käytetään usein sanaa ohjelmointikieli; tällä tarkoitetaan sekä merkintäkieliä että varsinaisia ohjelmointikieliä. Kummankin kielen tarkoituksena on kertoa sovellukselle miten toimia tai esittää asioita.
Responsiivinen sivusto	Sivuston sisältö ja näkyvyys muovautuu näyttökoon mukaan.
Sisällönhallintajärjestelmä	Tietojärjestelmä, joka palvelee koko organisaation sisällönhallintaa sen sijaan, että olisi keskittynyt pelkästään johonkin yksittäiseen osa-alueeseen, kuten verkkopalveluiden hallintaan.

2 Tietoperusta

Tässä luvussa käsitellään kaikkia niitä asioita, jotka liittyvät yksinkertaisen ja responsiivisen verkkopalvelun luomiseen ohjelmistokehityksen yleisten periaatteiden mukaisesti. Opinnäytetyön lähteinä käytettiin alan kirjallisuutta sekä verkkolähteitä. Käytetyt lähteet olivat sekä suomen- että englanninkielisiä.

2.1 Verkkopalvelun määritelmä

Verkkopalvelu eli palvelu joka on käytettävissä internetin kautta. Verkkopalvelu tarjoaa käyttäjälleen jotain tiettyä palvelua. Osa verkkosivuista tarjoaa pelkästään tietoa, mutta verkkopalvelu tarjoaa sekä tietoa että palvelua. Verkkopalveluita on julkisia ja yksityisiä. Julkiset verkkopalvelut ovat kaikkien käytettävissä, mutta yksityiset ovat vain tietyn organisaation tai ryhmän käytettävissä. Julkisia verkkopalveluita ovat muun muassa pankkien verkkopalvelut sekä verkkokaupat. Yksityisiä verkkopalveluita ovat muun muassa yritysten intranet sivut, joihin on pääsy vain yrityksen sisäverkosta. Olipa palvelut mitä tahansa mitä verkkopalvelu tarjoaa, niiden käyttö kuitenkin tuo lisäarvoa käyttäjälle. (Sinkkonen, Nuutila & Törmä 2009, 25-27.)

Nykyään verkossa on monenlaisia palveluita, joista osa on hyötypalveluita ja osa viihdepalveluita. Hyötypalveluita ovat muun muassa tiedon välittäminen sekä tallentaminen verkossa, kaupanteko verkossa, kommunikointi ja asiointi. Viihdepalveluihin lukeutuvat lukuisat verkkopelit, elokuvien katselu verkossa sekä musiikin kuuntelu verkossa. Tekniikan kehittyessä ja verkon nopeuksien parantuessa yhä enemmän hyöty- ja viihdepalveluita siirtyy verkkoon. Verkkopalveluiden käytöstä on tullut arkipäivää. (Sinkkonen ym. 2009, 25.)

Verkkopalveluiden sisältö voi myös vaihdella. Yleensä verkkopalveluiden sisältö dynaamista ja/tai staattista. Staattiselle sisällölle ominaista on sen samankaltaisuus kaikille käyttäjille ja sisältö voi pysyä samana pitkäänkin. Dynaaminen sisältö taas muuttuu käyttäjän mukaan aktiivisesti. Jos palvelu on aina toiminnallinen, tällöin puhutaan operatiivisesta palvelusta eli verkkosovelluksesta joka on interaktiivisempi ja monimutkaisempi toiminnoiltaan sekä arkkitehtuuriltaan kuin tavallinen verkkosivu. (Sinkkonen ym. 2009, 26-27.)

Vastuu verkossa olevasta sivustosta riippuu sen sisällöstä. Yleensä verkkosivuista vastaa yrityksen viestintävastaava ja verkkosovelluksista taas yrityksen tietohallinto. (Sinkkonen ym. 2009, 26.)

2.2 Käyttöliittymän suunnittelu

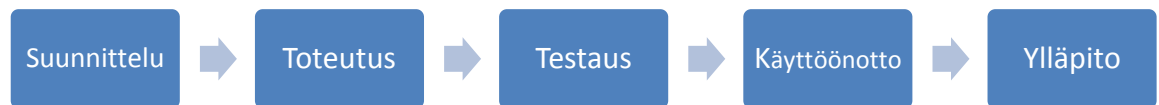
Palveluiden on oltava helposti lähestyttäviä sekä helppokäyttöisiä, jotta ne menestyisivät. Tähän voidaan päästä käyttäjäkeskeisellä suunnittelulla, jossa yhdistyy liiketoiminnan kannalta tärkeät tavoitteet sekä mitä käyttäjät tarvitsevat palvelulta. Käyttäjäkeskeisessä suunnittelussa oletettujen palvelun loppukäyttäjien käyttökokemus on tärkein ohjaava tekijä palvelun suunnittelussa. Siinä missä yrityksen tavoitteet ja käyttäjien tarpeet luovat tarpeen palvelulle, käyttökokemus ohjaa suunnittelua jotta palvelusta muodostuisi sellainen jota käyttäjät voivat käyttää vaivattomasti ja täten päästäisiin tavoitteisiin. (Sinkkonen ym. 2009, 27.)

Ihmisen ja koneen välisen interaktion tavoitteena on saada aikaan jotain. Käytettävyys on tärkeä osa tätä interaktiota, jotta tekeminen olisi tehokasta, vaivatonta ja tyydyttävää käyttäjälle. Käytettävyys muodostuu helppokäyttöisyydestä, tuottavuudesta, tehokkuudesta, vaivattomuudesta, opittavuudesta, muistettavuudesta sekä käyttäjätyytyväisyydestä. (Hartson & Pyla 2012, 10-11.)

Syitä käyttäjäkeskeiselle suunnittelulle on lukuisia, jotka jakaantuvat taloudellisiin hyötyihin, tehokkuuden parantamiseen sekä kilpailuedun saavuttamiseen. Käyttäjän ehdoilla ja hyvin suunniteltu palvelu sitouttaa käyttäjänsä ja luo tyytyväisyyttä, joka heijastuu palvelun suosituksina. Hyvin suunniteltu palvelu on myös laadukas, joka herättää luottamusta käyttäjässään. Palvelun ollessa suunniteltu käyttäjäkeskeisesti se myös tehostaa käyttäjän työskentelyä ja parantaa työhyvinvointia. Rahan satsaaminen kunnolliseen suunnitteluun maksaa itsensä takaisin, koska potentiaaliset asiakkaat yleensä hylkäävät palvelun jos se koetaan heti alkumetreillä hankalaksi. Etenkin jos palvelulla on kilpailijoita, käyttäjäkeskeisyydellä voidaan voittaa monta asiakkuutta ja se luo ammattilypeyttä. (Sinkkonen ym. 2009, 298-29.)

Käyttäjäkeskeisen suunnittelun menetelmiin kuuluu käyttäjien maailmaan perehtyminen sekä tutkiminen, käyttäjien käyttötarinat ja prototyypit. Jotta käyttäjien tarpeita voitaisiin ymmärtää, tulee palvelun suunnittelijoiden perehtyä käyttäjien toimintaympäristöön ja toimintamalleihin. Palvelun toimintaa todenmukaisesti mallintavat käyttäjien käyttötarinat, joiden pohjalta prototyyppi voidaan luoda. Prototyyppien ja niistä saatujen palautteiden pohjalta voidaan ohjata palvelun kehitystyötä oikeaan suuntaan. (Sinkkonen ym. 2009, 27.)

Käyttäjäkeskeinen suunnittelu alkaa konseptisuunnittelulla ja hyväksytty konsepti etenee toteutukseen. Konsepti otetaan käyttöön hyväksytyjen testausten jälkeen. Kun konsepti on virallisesti käyttöön otettu, voidaan puhua palvelusta, jota ylläpidetään ja jälkiseuraataan. Palvelun konseptointi on hyvin tärkeää heti alussa, koska konseptin avulla voidaan arvioida kustannuksia tehokkaammin kuin pelkällä suunnitelmalla. Samalla voidaan pohtia palvelun kannattavuutta ennen kuin projekti etenee suunnitteluvaiheeseen. Palvelun konsepti mallintaa palvelun ulkoista käyttöliittymää ja sen alla olevia kerroksia aletaan mallintamaan vasta kun projekti voidaan viedä toteutukseen. (Sinkkonen ym. 2009, 31.)



Kuvio 1. Kokonaisprosessi. (Sinkkonen ym. 2009, 31.)

Koko palvelun suunnittelun ja kehityksen pohjana toimii tietyt periaatteet. Näitä periaatteita ovat:

- Palvelun tulisi tukea käyttäjän luonnollista tapaa toimia ja olla helppokäyttöinen
- Palvelun navigoinnin pitää olla selkeä ja tehokas
- Palvelun suunnitteluratkaisuiden tulee olla johdonmukaisia ja jäljitettävissä
- Palvelun käyttämät käsitteet tulevat käyttäjien käsitemaailmasta
- Palvelun toiminnot vastaavat täsmällisesti tarpeeseen ja sille asetettuihin vaatimuksiin
- Palvelu opastaa käyttäjäänsä tarvittaessa tai sen on oltava niin intuitiivinen että opastusta ei tarvita
- Palvelun sisältö on relevanttia

(Sinkkonen ym. 2009, 35-37.)

2.3 Konseptin suunnittelu

Konseptin suunnittelulle tärkeää on käyttää jo heti tässä aikaisessa vaiheessa sellaisia kuvauksia, jotka mallintavat palvelun lopullista käyttäytymistä mahdollisimman tarkasti ja ymmärrettävästi. Tällöin kaikki ymmärtävät palvelun suunnitteluun liittyvät päätökset ja niiden vaikutukset. Konseptin suunnittelu palvelee myös jäljitettävyyttä. Kaikki lopullisen käyttöön otettavan palvelun toiminnallisuudet ovat jäljitettävissä konseptin suunnittelu vaiheeseen. (Sinkkonen ym. 2009, 33-34.)

Konseptin suunnittelu vaiheessa oleellisena menetelmänä on käyttötarinat eli user stories. Käyttötarinoita edeltää toimintatarinat, jotka kertovat miten nykyään toimitaan ja käyttötarinoilla taas kerrotaan miten tulevaa palvelua käytetään. Palvelun toiminnallisuudet on

tarkoitus rakentaa toimintatarinoiden innoittamina, oli palvelun toiminnallisuuden tarkoitus sitten digitalisoida olemassa oleva toiminta tai luoda jotain uutta joka tehostaa olemassa olevaa toimintaa. (Sinkkonen ym. 2009, 33.)

Toimintatarinoista voidaan tunnistaa käyttäjä ja toimintaympäristö, jonka pohjalta määritellään vaatimukset palvelulle. Tämän jälkeen laaditaan käyttötarinat, jotka ovat käytännössä toimintaratkaisuita palvelulle. Käyttötarinoiden mukaisia ratkaisuita verrataan vaatimuksia vasten ja jos ne täyttävät vaatimukset, voidaan käyttötarina siirtää osaksi käyttäjäkeskeistä suunnitelmaa joka etenee myöhemmin toteutukseen. (Sinkkonen ym. 2009, 34.)



Kuvio 2. Käyttäjäkeskeisen tuotekehityksen idea. (Sinkkonen ym. 2009, 34.)

2.3.1 Toimintatarinat

Toimintatarinoiden tarkoituksena on mallintaa nykyistä käyttäytymistä ja/tai toimintaa. Mallinnus tehdään kaavioksi, josta nähdään jokaisen asian syy-seuraus suhde. Alussa on alkutilanne, joka laukaisee toiminnan ja lopussa on lopputulos. Alkutilanteen ja lopputuloksen väliin muodostuu monen eri tekijän ja toiminnan askelmat. Kun mallinnus on tehty, voidaan toimintatarinasta löytää ongelmakohtat joihin palvelu voi vastata. Koska palveluiden kehittämisen tavoitteena on tehdä tekemisestä tehokkaampaa, on tarkoituksen mukaista pyrkiä automatisoimaan toimintatarinan mallista mahdollisimman monta manuaalista askelta sekä vähentää eri toimijoiden määrää. (Robertson & Robertson 2013, 94-97.)

2.3.2 Vaatimusmäärittely ja skenaariot

Vaatimusmäärittely alkaa skenaarioissa, joilla kerrotaan palvelun toimintojen haluttua toimintatapaa. Vaatimusmäärittelyä tapahtuu myös käyttötarinoissa, mutta niissä määritellään yksittäisiin toiminnallisuuksiin liittyvät vaatimukset. Vaatimusmäärittely on vaativa, mutta tärkeä osa palvelun suunnittelua. Vaatimusmäärittely ei välttämättä ole lista asioita, joita odotetaan palvelulta. Vaatimusmäärittely nivoutuu osaksi sekä skenaarioita että käyttötarinoita. Liiketoiminta asettaa aina omat vaatimuksensa palvelulle ja skenaarioissa

määritetyt vaatimukset tulee täsmätä liiketoiminnan vaatimusten ja tavoitteiden kanssa. (Robertson & Robertson 2013, 20 & 130.)

Skenaario on neutraali ja yksinkertainen tapa kertoa kaikille sidosryhmille, että miten palvelun toiminnan on tarkoitus toimia. Skenaariota muokataan kunnes se vastaa kaikkien sidosryhmien käsitystä palvelun odotetusta toiminnasta. Aivan kuten toimintatarinakin, skenaario mallinnetaan rikkomalla se askelmiin jotka edustavat toimijoita sekä tekijöitä. Alussa on alkutilanteen laukaiseva tekijä sekä lopussa lopputulos. Väliin muodostuu ne askeleet joilla palvelun toiminnallisuuden odotetaan saavuttavan lopputulos. (Robertson & Robertson 2013, 131 & 135-137.)

2.3.3 Käyttötarinat

Käyttötarinoiden tarkoitus on määrittää palvelun toiminnallisuuden toiminta. Käyttötarinoissa määritetään vaatimukset joiden mukaan toiminnallisuuden tulee toimia. Skenaariot toimivat pohjana käyttötarinoille ja yhdestä skenaarioista voi tulla useampikin käyttötarina. (Robertson & Robertson 2013, 329.) Käyttötarinoista käytetään yleisesti myös termiä käyttötäpaus, joka on kuitenkin sama asia. (Agile modeling, 2014.)

Käyttötarinat muodostuvat aina seuraavanlaisesti samalla kaavalla eli "Käyttäjänä X haluan toiminnon Y koska Z". Käyttäjä X määritetään, koska käyttötarinassa pitää tunnistaa toimija. Toiminto Y on itse toiminnallisuus, jota käyttötarinassa määritetään ja Z on perustelu sille miksi toiminto Y halutaan osaksi palvelua. Kaikki palvelun toiminnallisuudet pitää pystyä jäljittämään johonkin käyttötarinaan samalla tavalla kuin jokainen käyttötarina on jäljitettävissä skenaarioon. (Robertson & Robertson 2013, 331.)

Käyttötarinan nimi:	Esimerkki tarina
Määritelmä	Tarinan tarkoitus on kertoa ohjelmoijalle haluttu toiminnallisuus.
Perustelu	Jotta ohjelmoija tietää, mitä käyttäjä haluaa.
Lähde	Skenaario X
Onnistumisen määritelmä	Jos ohjelmoija ymmärtää tarinan perustelun ja määritelmän sekä ohjelmoida sen perusteella toiminnon
Prioriteetti	Liiketoiminta kriittinen toiminto
Käyttötarinan luoja	Matti Möttönen

Kuvio 3. Käyttötarinan mallipohja. (Robertson & Robertson 2013, 332.)

Käyttötarinat voidaan myös ilmentää käyttötapauskaaviona käyttötarinan tueksi. Tällöin toiminto on keskiössä ja sitä ympäröivät toimijat. Kaavioon piirretään toiminnon ja toimijoiden väliset suhteet. (Kasurinen 2013, 30-31.) Lopuksi palvelu toteutetaan käyttötarinoiden pohjalta ja testataan.

2.4 Testaaminen ja testauksen suunnittelu

Palvelu on testattava, jotta se toimisi odotetunlaisesti. Ohjelmistotestaus eli lyhyesti testaaminen on työtä, jolla varmistetaan että verkkopalvelu toimii kuten sen pitäisi ja että verkkopalvelu vastaa sille suunnitteluvaiheessa asetettuihin vaatimuksiin. Testaaminen on samalla verkkopalvelun kehityksen ohjaamista oikeaan suuntaan. (Kasurinen 2013,10.)

Testaaminen jakaantuu kolmeen osaan; yksikkötestaukseen, integrointitestaukseen ja järjestelmätestaukseen. Nämä kolme testauksen päävaihetta suoritetaan testi- ja kehitysympäristöissä. Testaaminen joka tapahtuu tuotantoympäristössä tai sitä simuloivassa ympäristössä kutsutaan hyväksymistestaukseksi, koska sen tarkoituksena on vain varmistaa palvelun toimivuus sen lopullisessa ympäristössä. (Kasurinen 2013,51.)

2.4.1 Yksikkötestaus

Yksikkötestauksessa testataan vain yhtä toiminnallisuutta tai komponenttia. Sen tarkoituksena on varmistaa, että kaikki toimii suunnitellusti yhden toiminnallisuuden kohdalla. Yksikkötestaus on testaamisen kaikkein tavallisin muoto ja siitä yleensä vastaa kehittäjä toiminnon vielä ollessa kehitysympäristössä. (Kasurinen 2013,51-54.)

Yksikkötestausta on esimerkiksi lomakkeen validoinnin tarkistaminen tai yhden raportoinnin muodostuminen ruudulle oikein. Yksikkötestauksen apuna voidaan käyttää funktioita tai ohjelmia, jotka esimerkiksi syöttävät toiminnolle tietoa tai kutsuvat toimintoa, jolloin voidaan varmistaa että palvelun toiminto reagoi oikein. (Kasurinen 2013,51-54.)

2.4.2 Integrointitestaus

Integrointitestaus on yksikkötestauksen jälkeen seuraava vaihe, kun yksittäisiä osia aletaan yhdistämään isommaksi kokonaisuudeksi. Integrointitestaus tehdään aina vähintään kahden eri komponentin tai toiminnallisuuden välillä. Integrointitestauksen tarkoituksena on varmistaa eri komponenttien ja toiminnallisuuksien välinen toimivuus. (Kasurinen 2013,54-56.)

Vaikka koko järjestelmä ei olisi valmis, useamman eri toiminnallisuuden välistä toimivuutta kuitenkin voidaan testata jos osat vain liittyvät toisiinsa. Integroititestausta on esimerkiksi se, että varmistetaan tiedon tallentuminen lomakkeelta tietokantaan tai että sisäänkirjautuminen palveluun onnistuu eritasoisilla käyttäjätunnuksilla. (Kasurinen 2013,54-56.)

Integroititestaus yleensä toteutetaan kehitysvaiheessa, kun uutta tai muutettua toiminnallisuutta halutaan testata suhteessa muihin järjestelmän osiin. Tähänkin löytyy monia funktioita ja ohjelmia, joiden avulla toiminnallisuuksia voidaan testata irrallaan itse palvelusta. (Kasurinen 2013,54-56.)

2.4.3 Järjestelmätestaus

Järjestelmätestaus on testauksen viimeinen vaihe. Tässä vaiheessa palvelu on testaustavalle valmis ja testaaminen suoritetaan sille tarkoitetussa testiympäristössä. Järjestelmätestauksen aikana testataan kaikkia järjestelmän toimintoja irrallaan ja yhdessä. Tässä vaiheessa testaamista yleensä suoritetaan valikoidun testaajan toimesta. Testaaja raportoi mahdolliset ongelmat suoraan kehittäjälle, joka voi korjata ongelmat. (Kasurinen 2013,56-57.)

Järjestelmätestausta varten laaditaan käyttötapauksista testitapaukset, jotka suoritetaan palvelulla. Testitapauksien tarkoituksena on varmistaa, että palvelun kaikki toiminnallisuudet toimivat kuten ne on suunniteltu. Testitapauksia ovat esimerkiksi käyttäjätunnusten luominen, raportin ajaminen järjestelmästä tai tietojen muuttaminen. (Kasurinen 2013,56-57.)

Testitapaukset voivat olla myös tavoitteellisia skenaarioita, joissa testaaja määrittää tietyt tavoitteet testitapauksella jotka testin aikana tulee saavuttaa. Tällöin yksi testitapaus voi testata useampaa toiminnallisuutta. Yksi esimerkki tällaisesta testitapauksesta on ensiksi varmistaa rekisteröitymisen onnistuminen, jonka jälkeen kokeillaan sisäänkirjautumista ja tietojen muuttamista. (Kasurinen 2013,56-57.)

2.5 Sisällönhallintajärjestelmät

Nykyään verkkosivuja voidaan rakentaa hyvin suosituiksi muodostuneilla sisällönhallintajärjestelmillä kuten WordPressillä tai Joomlailla. Sisällönhallintajärjestelmillä voidaan luoda ja hallinnoida erilaisia verkkoon julkaistavaa sisältöä kuten verkkosivuja ja blogikirjoituksia. (Vierityspalkki, 2009.)

Iso etu sisällönhallintajärjestelmien käytössä on niiden helppokäyttöisyys ja valmius tuottaa sivuja editorilla, sen sijaan että koodaisi kaiken käsin. Editorien avulla sivuille voidaan tuottaa tekstiä, kuvia ja videota.

Wordpressin kaltaisilla sisällönhallintajärjestelmillä voidaan luoda sivustoja monenlaisiin tarpeisiin kuten julkisia verkkosivuja, henkilökohtaisia blogisivuja ja intranet-sivustoja yritysten, yhteisöjen ja yksityishenkilöidenkin käyttöön. (Vierityspalkki, 2009.)

2.5.1 Wordpress

Wordpress on yksi monista sisällönhallintajärjestelmistä, jotka on rakennettu PHP/MySQL-pohjalle. Wordpressin avulla sisällöntuottajat voivat hallinnoida ja tuottaa sisältöä verkkosivuille verkossa toimivan selainpohjaisen käyttöliittymän kautta. Normaalisti sisältöä tuotetaan ja muokataan HTML-tiedostoihin jotka ladataan palvelimelle. (Leary 2010, kappale 1 sivu 1.)

Wordpress sai alkunsa 2003 ja se on sen jälkeen kasvanut hyvin laajalti käytetyksi sisällönhallintajärjestelmäksi. Wordpress on yksi maailman suosituimmista sisällönhallintajärjestelmistä. Sillä on laaja ja aktiivinen kehittäjäkunta, koska se on vapaan lähdekoodin projekti. Wordpress-sivut ovat täysin muokattavissa. (Wordpress, 2016.)

Wordpress on rakennettu PHP:llä ja MySQL:llä ja se on lisensoitu GPLv2 alle. Wordpress on hyvin joustava järjestelmä ja lukuisten pluginien avulla voi verkkosivuista tehdä halunsa kaltaisen. Plugins eli lisäosat ovat erilaisten organisaatioiden tai yksityishenkilöiden luomia lisätoiminnallisuuksia, joita voi liittää omaan Wordpress-sivustoon. Osa plugineista on ilmaisia ja osa maksullisia. Maksullisistakin plugineista on yleensä ilmainen versio tarjolla testausta varten. Tarvittaessa halutut muutokset voi koodata itse tai luoda tarvittaessa plugini. (Wordpress, 2016.)

Wordpressillä voidaan luoda yksittäinen sivusto tai useampi sivusto esimerkiksi koko konsernin kaikille yhtiölle. Tällöin puhutaan multisite-ratkaisusta. Erilaiset isännöintipalvelut eli hosting-palvelut tarjoavat suoraan mahdollisuuden luoda verkkosivut ja saada ne esille verkkoon maksua vastaan. Tällöin käyttäjän vastuulle on vain huolehtia maksusta ja sivujen luomisesta, jolloin hosting-palvelu huolehtii verkkosivujen näkymisestä verkossa. Verkkosivut ovat tällöin hosting-palvelun palvelimella. Toinen vaihtoehto on ylläpitää sivustoa omalta palvelimelta, jolloin ylläpitovastuu on täysin omalla vastuulla. (Wordpress, 2016.)

Wordpressissä on sisäänrakennettu tuki siistityille URL-osoitteille, mikroformaateille, kategorioille ja tunnisteille sekä standartoidut teemat. Nämä asiat helpottavat merkittävästi sivujen hakukoneoptimointia, joka on elintärkeää etenkin aloitteleville pienille yrityksille. (Leary 2010, kappale 1 sivu 4.)

3 Hoitoajat-verkkopalvelun vaatimusmäärittely

Tässä osiossa esitellään Hoitoajat-verkkopalvelun vaatimusmäärittely, jonka perusteella verkkopalvelu luotiin. Verkkopalvelun luominen tapahtui ohjelmistokehitykselle tyypillisen kehityskaaren läpi. Kaikki alkoi verkkopalvelun vaatimusmäärittelyllä, jossa määriteltiin ne tarpeet ja vaatimukset joihin verkkopalvelun tuli vastata. Toimintatarinalla kuvattiin nykyistä toimintaa johon tarkoitus löytää digitaalinen ratkaisu. Skenaariot kuvasivat tilanteita, joiden tilalle piti luoda toiminnallisuus verkkopalveluun. Skenaarioiden pohjalta luotiin käyttötarinat, jotka toimivat pohjana verkkopalvelun toiminnallisuuksien toteutukselle.

3.1 Verkkopalvelun yleiset vaatimukset

Tässä luvussa käydään läpi vaatimukset joihin verkkopalvelun piti vastata sekä käyttötapaukset joihin verkkopalvelun toiminnallisuudet piti vastata. Verkkopalvelun idea syntyi tarpeesta ilmoittaa lapsen hoitoajat verkon kautta.

Verkkopalvelun on toimittava yleisimmillä selaimilla, joita ovat Internet Explorer, Firefox, Google Chrome ja Safari. Verkkopalvelun ei ole tarkoitus olla yhteensopiva selaimien vanhempien versioiden kanssa.

Verkkopalvelulle oli asetettu seuraavat vaatimukset, jotka sen tuli myös täyttää.

Vaatus	Perustelu
Helppokäyttöisyys	Käyttäjäkunta molemmista käyttäryhmistä ei ole teknisesti orientoitunut
Responsiivinen	Jotta hoitoajat voi ilmoittaa myös mobiililaitteilla.
Tietoturallinen	Käyttäjien tunnukset on salattu ja palveluun eri kerätä paljoa henkilötietoja
Laajennettavissa	Verkkopalvelun prototyyppi on yksinkertainen, mutta sen pitää olla laajennettavissa uusilla toiminnallisuuksilla

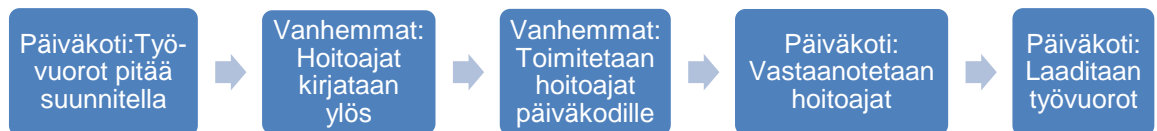
3.1.1 Toimintatarina nykytilasta

Lasten hoitoajat ilmoitetaan päiväkodille ainakin viikko etukäteen, jotta päiväkotitoi voi suunnitella työvuorot. Samalla lapulla pystyi halutessaan ilmoittamaan kaksi viikkoa etukäteen.

Hoitoajan ilmoitus koostui seuraavista tiedoista:

- Lapsen nimi
- Päivämäärä
- Alkuaika
- Nouto aika
- Muuta

Paperisen lapun toimittamisen sijaan tarkoituksena oli luoda verkkopalvelu jonka kautta samat tiedot voisi ilmoittaa.



Kuvio 4. Hoitoaikojen ilmoituksen toimintatarina.

3.1.2 Hoitoajat-verkkopalvelun skenaariot

Verkkopalvelussa on kaksi eri käyttäjäryhmää. Toinen on käyttäjät eli lasten vanhemmat, jotka ovat päiväkodin asiakkaita. Toinen taas on ylläpitöryhmä eli päiväkodin henkilökunta.

Lasten vanhemmat eli käyttäjät voivat kirjautua sisään ja ilmoittaa lasten hoitoajat. Ilmoitettuja hoitoaikoja ei voi muokata käyttäjän toimesta, koska päiväkotitoi suunnittelee työajat ilmoitettujen hoitoaikojen perusteella.

Päiväkodin henkilökunta eli ylläpitäjät voivat kirjautua sisään, tarkastella lasten hoitoaikoja sekä luoda uusia käyttäjiä.

Verkkopalvelu ei ole avoin eli siihen ei voi liittyä rekisteröitymällä julkisen verkkosivun kautta. Palveluun rekisteröityminen tapahtuu ylläpidon kautta.

3.1.3 Käyttötarinat

Vaatimusmäärittelyn käyttötarinataulukko:

Käyttötarina	Toimija
Hoitoaikojen ilmoittaminen	Lasten vanhemmat
Hoitoaikojen käsittely	Päiväkoti
Hoitoaikojen tarkastelu	Päiväkoti
Käyttäjätietojen ylläpito	Lasten vanhemmat ja päiväkoti

Käyttötarinan nimi:	Hoitoaikojen ilmoittaminen
Määritelmä	Vanhemmat voivat ilmoittaa lapsen hoitoajat etukäteen verkkopalvelun kautta.
Perustelu	Jotta paperisia lappuja ei enää tarvitse muistaa täyttää ja tiedot voisi ilmoittaa milloin vain, missä vain.
Lähde	Henri Komulainen
Onnistumisen määritelmä	Hoitoajat voi ilmoittaa onnistuneesti verkkopalvelun kautta.
Prioriteetti	Palvelukriittinen toiminto
Käyttötarinan luoja	Henri Komulainen

Käyttötarinan nimi:	Hoitoaikojen käsittely
Määritelmä	Päiväkoti voi hyväksyä tai hylätä ilmoitetut hoitoajat. Tapahtumasta lähtee vanhemmille ilmoitus sähköpostitse.
Perustelu	Jotta päiväkoti voi hallinnoida ilmoitettuja hoitoaikoja, ilmoitetut hoitoajat ei hyväksytä automaattisesti vaan ne pitää tarkistaa.
Lähde	Henri Komulainen
Onnistumisen määritelmä	Hoitoajat voi hyväksyä tai hylätä onnistuneesti verkkopalvelun kautta.
Prioriteetti	Palvelukriittinen toiminto
Käyttötarinan luoja	Henri Komulainen

Käyttötarinan nimi:	Hoitoaikojen tarkastelu
Määritelmä	Päiväkoti voi tarkastella sekä ilmoitettuja että hyväksytyjä hoitoaikoja verkkopalvelun kautta.
Perustelu	Jotta päiväkodin henkilökunnan työajat voisi suunnitella.
Lähde	Henri Komulainen
Onnistumisen määritelmä	Hoitoajat voi tarkistaa onnistuneesti verkkopalvelun kautta.
Prioriteetti	Palvelukriittinen toiminto
Käyttötarinan luoja	Henri Komulainen

Käyttötarinan nimi:	Käyttäjätietojen ylläpito
Määritelmä	Vanhemmat voivat päivittää tietoja, mutta vain päiväkotia voi rekisteröidä uusia käyttäjiä.
Perustelu	Jotta käyttäjät voisivat itse ylläpitää omia tietojensa ajan tasalla ja päiväkodin hoitoajat järjestelmään ei voisi rekisteröityä kuka tahansa.
Lähde	Henri Komulainen
Onnistumisen määritelmä	Käyttäjätiedot voi onnistuneesti päivittää verkkopalvelun kautta. Uudet käyttäjätunnukset voi luoda onnistuneesti verkkopalvelun kautta.
Prioriteetti	Palvelukriittinen toiminto
Käyttötarinan luoja	Henri Komulainen

4 Verkkopalvelun luominen sisällönhallintajärjestelmällä

Hoitoajat-verkkopalvelu toteutettiin Wordpress-sisällönhallintajärjestelmällä. Käyttötapausten pohjalta verkkopalveluun rakennettiin niitä vastaavat toiminnallisuudet. Yksikkötestausta suoritettiin läpi verkkopalvelun rakentamisen, koska jokainen toiminto piti testata erikseen ennen integraatiotestausta. Integraatio- ja järjestelmätestaus suoritettiin sen jälkeen kun verkkopalvelu oli valmis.

Wordpressiin valintaan liittyi monia syitä, joita olivat:

- Laaja käyttäjäkunta
- Helppo käyttää
- Hyvin muokattavissa omiin tarpeisiin
- Alustan ja plugineiden ylläpito ja tietoturva on ison yhteisön harteilla
- Luodut Wordpress-sivustot on kopioitavissa
- Wordpressillä saa luotua responsiiviset verkkosivut
- Verkkosivut helpommin laajennettavissa

4.1 Hoitoajat-verkkopalvelu

Wordpressillä saatiin aikaiseksi verkkopalvelun julkinen ilme näyttöpäätteelle. Tarkoituksena oli luoda verkkopalvelu, joka on responsiivinen eli verkkosivusto muovautuu käyttäjän näyttöpäätteen mukaan. Kaiken olisi myöskin voinut koodata alusta alkaen itse, mutta sisällönhallintajärjestelmät ja ohjelmistokehykset ovat tehneet verkkosivujen rakentamisesta nopeaa ja helppoa. Wordpressin teemat ovat lähtökohtaisesti responsiivisia, mikä helpotti verkkopalvelun rakentamista.

Hoitoajat-verkkopalvelulle luotiin kaksi käyttäjäryhmää; käyttäjät ja pääkäyttäjät. Käyttäjät ovat lasten vanhemmat joille on luotu tunnukset verkkopalveluun ja pääkäyttäjät ovat päiväkotien henkilökunta.

Käyttäjät voivat kirjautua sisään palveluun, ilmoittaa lasten hoitoajat sekä muokata omia tietoja. Kun hoitoajat ilmoittaa, lähtee sähköposti-ilmoitus pääkäyttäjälle jotta hän tietää että hoitoaikoja on ilmoitettu. Hoitoajat ilmoitetaan merkitsemällä päivät kalenteriin ja täyttämällä lomakkeeseen saapumisaika, noutaika, lapsen nimi, sähköposti sekä mahdolliset lisätiedot.

Pääkäyttäjät voivat kirjautua hallinnointipaneeliin, mistä käsin he voivat luoda uusia käyttäjätunnuksia, muokata tai poistaa käyttäjätunnuksia sekä hallinnoida hoitoaikakalenteria ja sen ilmoituksia. Tarvittaessa pääkäyttäjät voivat muokata sivujen sisältöä.

Hoitoajat-verkkopalvelun kuvaus:

Sivu/Toiminto	Näkyvyys	Toiminto	Käyttäjä
Sisäänkirjautumissivu	Julkinen	Käyttäjien sisäänkirjautumissivu	Käyttäjät
Käyttäjien kalenterisivu (sisäänkirjautuminen kirjautumissivun kautta ohjaa suoraan tänne)	Vain käyttäjille ja pääkäyttäjälle	Lasten hoitoaikojen ilmoittamiseen tarkoitettu sivu. Sisältää kalenterin ja lomakkeen hoitoaikojen ilmoittamiseen sekä linkin omien tietojen muokkaamiseen.	Käyttäjät
Omat tiedot-sivu	Vain käyttäjille ja pääkäyttäjälle	Käyttäjien omien tietojen muokkaamiseen tarkoitettu sivu.	Käyttäjät
Käyttäjien rekisteröinti	Pääkäyttäjien hallinointipaneeli	Uusien käyttäjien rekisteröinti	Pääkäyttäjät
Käyttäjien hallinointi	Pääkäyttäjien hallinointipaneeli	Käyttäjätietojen muuttaminen ja poistaminen	Pääkäyttäjät
Ilmoitetut hoitoajat-paneeli	Pääkäyttäjien hallinointipaneeli	Käyttäjien ilmoittamat hoitoajat näkyvät listauksena, josta ne voi hyväksyä tai hylätä..	Pääkäyttäjät

4.2 Hoitoajat-verkkopalvelun toiminnot

Tässä kappaleessa on esitelty kaikki verkkopalvelun toiminnot, joita käyttäjät ja pääkäyttäjät voivat suorittaa. Käyttötarinat toimivat verkkopalvelun toimintojen laatimisen perustana. Toimintojen kuvaamisessa olennaisinta oli tunnistaa toimija tai toimijat, toiminnan esiehto sekä lopputulos. Toimintojen kuvauksissa oli myös lyhyesti kuvattu toiminnallisuuden toimintaperiaatteet.

4.2.1 Sisäänkirjautuminen

Toimijat	Käyttäjä/pääkäyttäjä
Esiehto	Toimijalla on tunnukset
Lopputulos	Toimija pääsee kirjautumaan sisään

Käyttäjä ja pääkäyttäjä voivat kirjautua sisään sisäänkirjautumissivun kautta. Tunnusten taso määrittelee näkyvyyden sisäänkirjautumisen jälkeen. Pääkäyttäjillä on mahdollisuus mennä suoraan hallinnointipaneelin, mutta käyttäjillä näkyy vain heille tarkoitettu kalenterisivu.

4.2.2 Hoitoaikojen ilmoittaminen

Toimijat	Käyttäjä
Esiehto	Toimija on kirjautunut sisään
Lopputulos	Ilmoitetut hoitoajat

Käyttäjä voi merkata kalenteriin päivät, joita ilmoitus koskee. Tämän jälkeen käyttäjä täyttää lomakkeen, jonka tietokenttiä ovat:

- Lapsen nimi
- Alkuaika
- Sähköposti
- Nouto aika

Kun tiedot lähettää, pääkäyttäjä saa sähköpostin ilmoitetuista hoitoajoista sekä kehoituksen käsitellä ilmoitus.

4.2.3 Omien tietojen muokkaus

Toimijat	Käyttäjä
Esiehto	Toimija on kirjautunut sisään
Lopputulos	Tietojen muokkaus tapahtunut onnistuneesti

Käyttäjä voi sisäänkirjautumisen jälkeen mennä omiin tietoihin linkin kautta, josta hän voi päivittää omat tietonsa.

4.2.4 Käyttäjien rekisteröinti

Toimijat	Pääkäyttäjä
Esiehto	Toimija on kirjautunut sisään ja on pääkäyttäjä oikeudet
Lopputulos	Uuden käyttäjän rekisteröinti

Pääkäyttäjä voi hallinnointipaneelin kautta rekisteröidä uusia käyttäjiä. Vain pääkäyttäjä voi rekisteröidä uusia käyttäjiä, jottei kukaan palvelun piiriin kuulumaton pääsisi syöttämään tietoja.

4.2.5 Käyttäjien hallinnointi

Toimijat	Pääkäyttäjä
Esiehto	Toimija on kirjautunut sisään ja on pääkäyttäjä oikeudet
Lopputulos	Pääkäyttäjä voi muokata käyttäjän tietoja tai poistaa käyttäjän kokonaan

Pääkäyttäjä voi hallinnointipaneelin kautta muokata käyttäjän tietoja tai poistaa käyttäjän kokonaan, jos palvelusopimus päättyy.

4.2.6 Ilmoitettujen hoitoaikojen käsittely

Toimijat	Pääkäyttäjä
Esiehto	Toimija on kirjautunut sisään ja on pääkäyttäjä oikeudet
Lopputulokset	Ilmoitettujen hoitoaikojen hyväksyminen tai hylkääminen

Pääkäyttäjä voi hyväksyä tai hylätä ilmoitetut hoitoajat. Jos tiedot ovat oikein, pääkäyttäjä voi hyväksyä hoitoajat. Jos tiedot ovat virheelliset, pääkäyttäjä voi hylätä ilmoitetut hoitoajat. Kummassakin tapauksessa käyttäjä saa asiasta sähköpostin.

4.2.7 Ilmoitettujen ja hyväksyttyjen hoitoaikojen tarkastelu

Toimijat	Pääkäyttäjä
Esiehto	Toimija on kirjautunut sisään ja on pääkäyttäjä oikeudet
Lopputulokset	Ilmoitettujen ja hyväksyttyjen hoitoaikojen tarkastelu

Pääkäyttäjä voi tarkastella ilmoitettuja ja hyväksyttyjä hoitoaikoja verkkopalvelun kautta, jotta päiväkotitoiminta voisi suunnitella työvuorot.

5 Hoitoajat-verkkopalvelu toteutus

Tässä osiossa käydään läpi lisäosat, jotka otettiin käyttöön Wordpressissä jotta verkkopalveluun saatiin halutut toiminnallisuudet. Tässä osiossa myös esitellään toteutetun verkkopalvelun sivut.

5.1 Käytetyt lisäosat

Simple Wordpress Membership, jota käytettiin käyttäjien ja käyttäjätasojen luomiseen. Käyttäjätasot suodattavat sivujen näkyvyyttä, joka toteutui sivuilla niin että käyttäjillä oli pääsy vain omalla käyttäjäkohtaiselle etusivulle sekä omiin tietoihin.

Simple Membership After Login Redirection lisäosa huolehti sisäänkirjautumisen jälkeisestä uudelleenohjauksesta. Lisäosaa käytettiin peruskäyttäjien uudelleen ohjaukseen sisäänkirjautumissivulta käyttäjäkohtaiselle etusivulle.

Modular Custom CSS, jota käytettiin sisäänkirjautumissivulla poistamaan ylimääräiset Join us- ja Registration-linkit.

Varauskalenteri eli Booking Calendar lisäosa, jota käytettiin varauskalenteri näkymien toteuttamiseen käyttäjien omille sivuille sekä pääkäyttäjille. Käyttäjille kalenteri näytti vapaat päivät, joille pystyi ilmoittamaan hoitoajat. Pääkäyttäjät pystyivät tarkastelemaan ilmoitettuja hoitoaikoja listana tai kalenterista.

5.2 Sisäänkirjautumissivu

Sisäänkirjautumissivun kautta käyttäjät voivat kirjautua sisään. Jos tunnukset ovat vanhempien, tunnuksilla avautuu pelkästään heidän oma sivunsa. Pääkäyttäjillä näkyy ohjauspaneeli. Jos käyttäjä on unohtanut salasanaan, hän voi tiedustella niitä 'Forgot Password'-linkin kautta, joka vie unohtunut salana-sivulle.



Tervetuloa Hoitoajat-verkkopalveluun

Kirjauduthan sisään palveluun.

Jos sinulla ei ole tunnuksia, olethan yhteydessä päiväkodin henkilökuntaan.

Username

Password

☐ Remember Me

LOGIN

[Forgot Password?](#)

Kuva 1. Etusivu

5.3 Unohtunut salasana

Tämän sivun kautta rekisteröityneet käyttäjät voivat lähettää sähköpostiinsa salasanan vaihtolinkin antamalla sähköpostinsa, jolla he ovat rekisteröityneet.



Password Reset

Email Address

RESET PASSWORD

Kuva 2. Unohtunut salasana.

5.4 Käyttäjän etusivu

Käyttäjille avautuu sisäänkirjautumisen jälkeen oma etusivu, josta he voivat ilmoittaa hoitoajat merkitsemällä päivät kalenteriin, joita ilmoitus koskee. Lomakkeelle käyttäjät täyttävät tiedot hoitoajoista eli milloin lapsi tuodaan hoitoon, milloin haetaan, lapsen nimi, sähköposti sekä muut tiedot. Kuittaus ilmoituksen hyväksymisestä tai hylkäämisestä tulee sähköpostiin. Sivun kautta voi päästä myös muokkaamaan omia tietojaan.

Käyttäjän etusivu

Voit muokata tietojasi [täältä](#).

Hoitoaikojen ilmoitus

Ilmoita hoitoajat kalenterin avulla.

Merkitse päivä(t), syötä aloitus- ja noutoajat sekä lisätiedot.

Saat sähköpostitse vahvistuksen hoitoajoista.

Toukokuu 2016

Kesäkuu 2016 >>

MA	TI	KE	TO	PE	LA	SU	MA	TI	KE	TO	PE	LA	SU
						1			1	2	3	4	5
2	3	4	5	6	7	8	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18	19
16	17	18	19	20	21	22	20	21	22	23	24	25	26
23	24	25	26	27	28	29	27	28	29	30			
30	31												

Powered by [WP Booking Calendar](#)

Nimi*:

Saapumisaika*:

Email*:

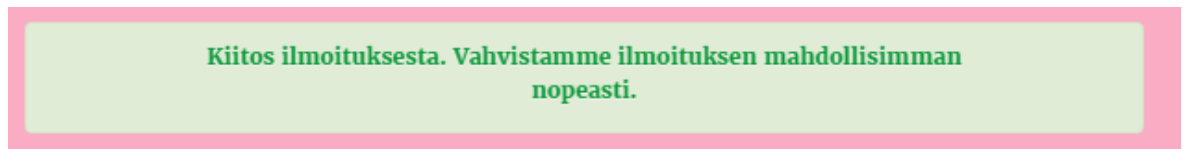
Noutaika*:

Lisätiedot:

LÄHETÄ

Kuva 3. Käyttäjän etusivu.

5.5 Kuittaus lähetetyistä hoitoajoista



Kuva 4. Kuittaus ilmoituksesta.

5.6 Käyttäjän tiedot-sivu

Käyttäjän tiedot-sivulla käyttäjä voi muokata omia tietojaan.

A screenshot of a web page titled "Profile" in a large, bold, black font. Below the title is a form with two columns. The left column contains labels for various user attributes, and the right column contains input fields or values. The attributes and their corresponding values are: Username (Henri), Email (komulainen.henri+1@gmail.com), Password (empty), Repeat Password (empty), First Name (Testi), Last Name (Henri), Phone (empty), Street (empty), City (empty), State (empty), Zipcode (empty), Country (empty), and Membership Level (Käyttäjä). At the bottom of the form is a black button with the word "UPDATE" in white. Below the button is a blue link labeled "Paluu".

Username	Henri
Email	komulainen.henri+1@gmail.com
Password	
Repeat Password	
First Name	Testi
Last Name	Henri
Phone	
Street	
City	
State	
Zipcode	
Country	
Membership Level	Käyttäjä

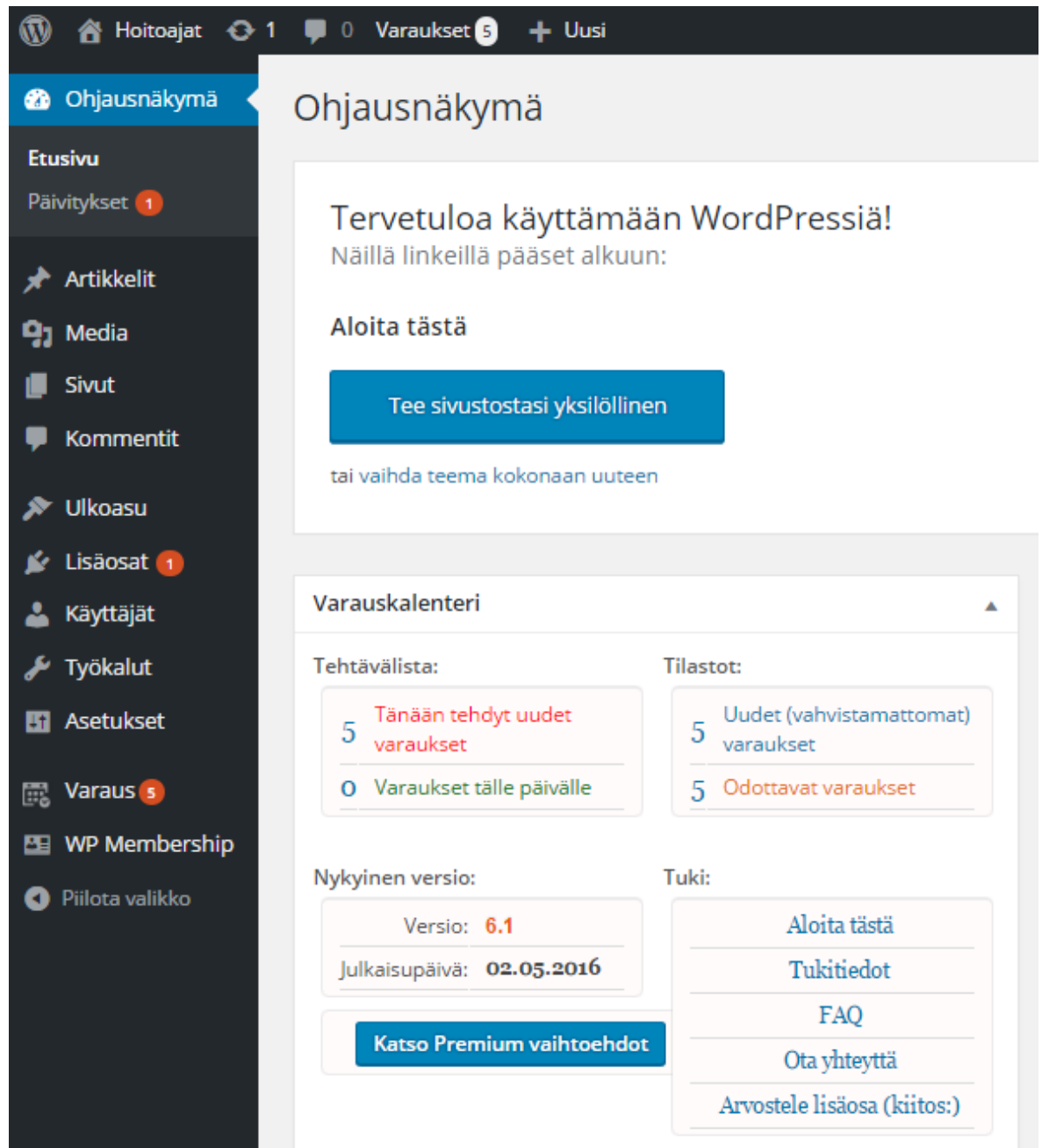
[Paluu](#)

UPDATE

Kuva 5. Käyttäjän tiedot.

5.7 Pääkäyttäjän etusivu

Pääkäyttäjän etusivulta pääkäyttäjä näkee heti, että onko vanhemmat lähettäneet uusia ilmoituksia. Etusivulta pääsee myös muokkaamaan sivujen sisältöä kohdasta Sivut.



The screenshot shows the WordPress dashboard for a user with the 'Ohjausnäkö' role. The top navigation bar includes links for 'Hoitoajat', '1', '0', 'Varaukset 5', and '+ Uusi'. The sidebar on the left contains a list of navigation links: 'Ohjausnäkö', 'Etusivu', 'Päivitykset 1', 'Artikkelit', 'Media', 'Sivut', 'Kommentit', 'Ulkoasu', 'Lisäosat 1', 'Käyttäjät', 'Työkalut', 'Asetukset', 'Varaus 5', 'WP Membership', and 'Piilota valikko'. The main content area is titled 'Ohjausnäkö' and features a welcome message: 'Tervetuloa käyttämään WordPressiä! Näillä linkeillä pääset alkuun:'. Below this is a button 'Aloita tästä' and a large blue button 'Tee sivustostasi yksilöllinen'. A link 'tai vaihda tema kokonaan uuteen' is also present. The 'Varauskalenteri' widget shows a 'Tehtävälista' with '5 Tänään tehdyt uudet varaukset' and '0 Varaukset tälle päivälle'. The 'Tilastot' widget shows '5 Uudet (vahvistamattomat) varaukset' and '5 Odottavat varaukset'. The 'Nykyinen versio' widget shows 'Versio: 6.1' and 'Julkaisupäivä: 02.05.2016', with a button 'Katso Premium vaihtoehdot'. The 'Tuki' widget lists links: 'Aloita tästä', 'Tukitiedot', 'FAQ', 'Ota yhteyttä', and 'Arvostele lisäosa (kiitos:)'.

Kuva 6. Pääkäyttäjän etusivu.

5.8 Pääkäyttäjän kalenteri näkymä

Kalenterinäkymässä pääkäyttäjä voi nähdä ilmoitetut hoitoajat listana tai kalenterina.

Suodatin

Toiminnot

Päivä

Tila: Kaikki

Päivämäärä: Valitut päivät

Järjestä: Päivämäärä

Lähetä sähköpostia

ID	Otsikko	Varauksen tiedot		Varatut päivät	Toiminnot
<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>
<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>
<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>
<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>
<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>
<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>

Kuva 7. Varaukset listana.

Päivämäärät	Maanantai	Tiistai	Keskiviikko	Torstai	Perjantai	Lauantai	Sunnuntai
touko 9, 2016 - touko 15, 2016							
touko 16, 2016 - touko 22, 2016							
touko 23, 2016 - touko 29, 2016							
touko 30, 2016 - kesä 5, 2016							
kesä 6, 2016 - kesä 12, 2016							
kesä 13, 2016 - kesä 19, 2016							
kesä 20, 2016 - kesä 26, 2016							
kesä 27, 2016 - heinä 3, 2016							
heinä 4, 2016 - heinä 10, 2016							
heinä 11, 2016 - heinä 17, 2016							
heinä 18, 2016 - heinä 24, 2016							
heinä 25, 2016 - heinä 31, 2016							

Kuva 8. Varaukset kalenterissa

5.9 Pääkäyttäjän näkymä käyttäjiin

Pääkäyttäjän käyttäjänäkymästä pääkäyttäjä voi muokata käyttäjien tietoja tai luoda uuden käyttäjän.

ID	Username	First Name	Last Name	Email	Membership Level	Access Starts	Account State
4	Henri	Testi	Henri	komulainen.henri+1@gmail.com	Käyttäjä	2016-05-02	active
3	Hootajat_admin	Komulainen	Henri	komulainen.henri@gmail.com	Pääkäyttäjä	2016-05-02	active
2	Testadmin	Testi	Admin	asfa@fff.com	Pääkäyttäjä	2016-05-01	active
1	Testaja	Testaja	Esa	asfa@fasafa.com	Käyttäjä	2016-05-01	active

Kuva 9. Käyttäjänäkymä.

5.10 Uuden käyttäjän lisääminen

Add Member

Create a brand new user and add it to this site.

Username *(required)*

E-mail *(required)*

Password *(twice, required)*

Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers and symbols like ! " ? \$ % ^ &).

Account Status

Active ▾

Membership Level

Pääkäyttäjä ▾

Access Starts

2016-05-14

First Name

Last Name

Gender

Not Specified ▾

Phone

Street

City

State

Zipcode

Country

Company

Member Since

2016-05-14

Add New Member

Kuva 10. Käyttäjän lisääminen.

6 Testaaminen

Verkkopalvelun testaaminen suoritettiin manuaalisesti, jotta mahdolliset käyttöliittymäongelmat huomattaisiin. Testaamisen suorittamiseen ei ollut soveliasta käyttää mitään automatisoitua sovellusta, koska verkkopalvelu on hyvin pieni ja yksinkertainen.

6.1 Yksikkötestaus

Mitään erillistä yksikkötestausta ei tarvinnut suorittaa, koska jokaisen palveluun lisätyn osan yksikkötestaus on suoritettu jo tekijöiden toimesta. Yksikkötestauksessa yleiset testattavat asiat eli tietojen tallentuminen, validointi ja nappien sekä linkkien toiminta on suoritettu ja varmistettu toimivaksi integrointitestauksen yhteydessä.

6.2 Integrointitestaus

Integrointitestaus suoritettiin kaikilla järjestelmän osilla, joita ovat:

1. Rekisteröityminen pääkäyttäjän näkymästä
2. Sisäänkirjautuminen
3. Tietojen syöttäminen ja tallentaminen käyttäjän näkymästä
4. Tietojen tarkastelu ylläpidon näkymästä

Sähköpostien testaamista ei voitu suorittaa, koska sähköpostien lähettäminen olisi vaatinut erillisen sähköpostipalvelimen vuokraamisen. Täten unohtuneen salasanan vaihtamista sekä hoitoaikojen hyväksymis/hylkäys-kuittauksia ei voitu testata.

6.2.1 Testitapaukset integrointitestaukselle

Testattava yksikkö	Testitapaus	Odotettu tulos	Lopputulos
Sisäänkirjautuminen	Käyttäjä kirjoittaa oikein tunnuksen ja salasanan	Käyttäjä siirtyy käyttäjän etusivulle	Käyttäjä siirtyi onnistuneesti omalle etusivulle
Sisäänkirjautuminen	Käyttäjä kirjoittaa väärintunnuksen ja salasanan	Sisäänkirjautuminen epäonnistuu ja käyttäjä saa kehoituksen tarkistaa annetut tiedot	Palvelu toimi oikein eli ei päästänyt käyttäjää sisään ja kehoitti tarkistamaan tiedot.
Kalenteri	Käyttäjä ilmoittaa hoitoajat merkitsemällä päivät kalente-	Käyttäjän antamat tiedot siirtyvät pääkäyttäjän kalenteriin tarkas-	Tiedot siirtyivät onnistuneesti

	riin ja täyttämällä lomakkeen	tusta varten	
Kalenteri	Pääkäyttäjä hyväksyy tai hylkää hoitoajat	Ilmoitetut hoitoajat saavat asianmukaisen merkinnän, vahvistettu tai hylätty	Tiedot tallentuvat oikein
Käyttäjätiedot	Käyttäjä päivittää tiedot omista tiedoistaan	Käyttäjän tiedot päivittyvät	Tiedot päivittyivät
Käyttäjätiedot	Pääkäyttäjä päivittää käyttäjän tiedot	Käyttäjän tiedot päivittyvät	Tiedot päivittyivät
Käyttäjätiedot	Pääkäyttäjä rekisteröi uuden käyttäjän	Käyttäjän tiedot luodaan onnistuneesti	Uusi käyttäjä luotiin onnistuneesti

6.3 Järjestelmätestaus

Järjestelmätestaukseen sovellettiin skenaarioita. Jokaisessa skenaariossa oli useita eri tapahtumia, joilla testattiin järjestelmän yleinen toimivuus. Verkkopalvelun skaalautuvuus näyttöpäätteen mukaan testattiin kokeilemalla palvelua eri kokoisilla näytöillä.

Testaaja	Skenaarion kuvaus	Odotettu tulos	Lopputulos
Käyttäjä	Käyttäjä kirjautuu sisään, syöttää ja lähettää hoitoajat.	Sisäänkirjautuminen tapahtuu onnistuneesti ja tiedot tallentuvat.	Kaikki toimi odotetusti.
Käyttäjä	Käyttäjä kirjautuu sisään ja muokkaa omia tietojaan.	Sisäänkirjautuminen tapahtuu onnistuneesti ja tiedot tallentuvat.	Kaikki toimi odotetusti.
Pääkäyttäjä	Pääkäyttäjä kirjautuu sisään ja hyväksyy ilmoitetut hoitoajat.	Sisäänkirjautuminen tapahtuu onnistuneesti ja tiedot tallentuvat.	Kaikki toimi odotetusti.
Pääkäyttäjä	Pääkäyttäjä kirjautuu sisään ja muokkaa käyttäjän tietojaan.	Sisäänkirjautuminen tapahtuu onnistuneesti ja tiedot tallentuvat.	Kaikki toimi odotetusti.

Pohdinta

Hoitoajat-verkkopalvelu toimi odotetunlaisesti, juuri niin kuin se oli suunniteltu ja toteutettu. Verkkopalvelun kautta käyttäjät pystyivät ilmoittamaan hoitoajat, joita pääkäyttäjät eli päiväkodin henkilökunta pystyi tarkastelemaan. Hoitoajat-verkkopalvelu vastasi sille laadittuihin vaatimuksiin eli se on responsiivinen, tietoturvallinen, helppo käyttää sekä hyvin laajennettavissa. Wordpressissä on vakiona käyttäjien salasanojen tiivistäminen ja suolaus, joka takaa salasanat on hyvin vaikea murtaa. Verkkopalvelun rakentaminen oli pitkä prosessi, koska sen toteutus piti miettiä tarkkaan. Jos olisin koodannut kaikki osat itse, aikaa olisi mennyt paljon enemmän ja kaikki osat olisi jäänyt ylläpidettäväksi itselleni. Tämä ei olisi palvellut verkkopalvelun tulevaisuuden suunnitelmia, koska verkkopalvelun tarkoituksena on olla myös helposti laajennettavissa uusilla ominaisuuksilla. Koska palvelu on toteutettu Wordpressillä, se on helposti replikoitavissa ja täten tuotteistettavissa.

Suomessa ei ole vastaavanlaisia palveluita päiväkodeille, jotka olisi yhtä helposti muokattavissa sekä responsiivisia. Jatkokehityksen kohteita olisi selkeästi palvelun lisäarvon kasvattaminen uusilla toiminnallisuuksilla sekä palvelun kielen yhdenmukaistaminen suomeksi. Osassa toiminnallisuuksista kieli on suomeksi, mutta esimerkiksi käyttäjien hallintapuoli jäi englannin-kieliseksi. Ajan rajallisuuden takia jätin kielen yhdenmukaistamisen pois opinnäytetyön toteutuksesta.

Toteutettu Hoitoajat-verkkopalvelu on myös tuotteistettavissa. Yksi tapa myydä tuotetta olisi tehdä siitä skaalautuva SaaS-palvelu eli asiakkaat maksaisivat vain käytön mukaan. Yksi esimerkki toteutus tästä olisi veloittaa rekisteröityjen käyttäjien määrän mukaan. Tällöin palvelun hinta olisi lähtökohtaisesti edullinen ja palvelun piiriin voitaisiin saada myös pienemmät päiväkodit. Konsernit, joihin kuuluu useampi päiväkotit, voisivat saada alennusta perustettujen päiväkotikohtaisten sivujen määrästä, mutta veloitus käyttäjien määrästä pysyisi samana. Lähtökohta on kuitenkin se, että jokaisella päiväkodilla on omat palvelusivunsa, joihin käyttäjät voisivat kirjautua sisään ja joihin päiväkodin henkilökunta voisi syöttää sisältöä.

Opin paljon opinnäytetyön aikana verkkopalvelun toteuttamisesta. Tietopohja loi perustan ammattimaiselle verkkopalvelukehitykselle ja muovasi samalla myynnillistä ajattelutapaa palveluita kehittäessä. Tarkoituksena ei ollut luoda mitään huvia vuoksi vaan luoda palvelukonsepti jota voisi myydä jatkokehityksen jälkeen. Nykyisessä muodossaan verkkopalvelu on liian pieni myytäväksi, mutta siinä on potentiaalia kasvaa. Verkkopalvelun toteuttamisen yhteydessä opin verkkopalveluiden luomisen olemassa olevilla sisällönhallintajärjestelmillä, mistä ei ole ollut aiempaa kokemusta.

Lähteet

Agile modeling. Luettavissa: <http://www.agilemodeling.com/> Luettu: 29.4.2016

Leary, S. 2010. Beginning Wordpress 3. 1. painos. Springer Science+Business Media, LLC. New York, NY.

Hartson, R. & Pyla, P. 2012. The UX Book. 1. painos. Elsevier, Inc. Waltham, MA.

Kasurinen, J. 2013. Ohjelmistotestauksen käsikirja. 1. painos. Docendo. Jyväskylä.

Robertson, J. & Robertson, S. 2013. Mastering the requirements process. 3. painos. Addison-Wesley. Upper Saddle River, NJ.

Sinkkonen, I., Nuutila, E. & Törmä S. 2009. Helppokäyttöisen verkkopalvelun suunnittelu. 1. painos. Tietosanoma Oy. Helsinki.

Vierityspalkki 2009. Käsitesekamelskaa: julkaisujärjestelmä, CMS, portaali, sisällönhallintajärjestelmä . Luettavissa: <http://vierityspalkki.fi/2009/11/03/kasitesekamelskaa-julkaisujarjestelma-cms-portaali-sisallönhallintajarjestelma/> Luettu: 25.4.2016

Wordpress. Luettavissa: <https://wordpress.org/> Luettu: 25.4.2016